struct s\_l3 {

uint8\_t CT\_RAMPON : 4; // 0x01.0, s:4 d: X GE COND\_VALUE\_LO

uint8\_t CT\_RAMPOFF : 4; // 0x01.4, s:4 d: X GE COND\_VALUE\_

uint8\_t CT\_ONDELAY : 4; // 0x02.0, s:4 d: X GE COND\_VALUE\_

uint8\_t CT\_OFFDELAY : 4; // 0x02.4, s:4 d: X GE COND\_VALUE\_

uint8\_t CT\_ON : 4; // 0x03.0, s:4 d: X GE COND\_VALUE\_

uint8\_t CT\_OFF : 4; // 0x03.4, s:4 d: X GE COND\_VALUE\_

uint8\_t COND\_VALUE\_LO : 8; // 0x04.0, s:8 d:

uint8\_t COND\_VALUE\_HI : 8; // 0x05.0, s:8 d:

uint8\_t ONDELAY\_TIME : 8; // 0x06.0, s:8 d: 0

uint8\_t ON\_TIME : 8; // 0x07.0, s:8 d: 111600.0

uint8\_t OFFDELAY\_TIME : 8; // 0x08.0, s:8 d: 0 s

uint8\_t OFF\_TIME : 8; // 0x09.0, s:8 d: 111600.0 s

uint8\_t ACTION\_TYPE : 4; // 0x0a.0, s:4 d: INACTIVE

uint8\_t : 1; // 0x0a.4, s:2 d:

uint8\_t MULTIEXECUTE : 1; //

uint8\_t OFF\_TIME\_MODE : 1; // 0x0a.6, s:1 d: ABSOLUTE

uint8\_t ON\_TIME\_MODE : 1; // 0x0a.7, s:1 d: ABSOLUTE

uint8\_t JT\_ON : 4; // 0x0b.0, s:4 d: ONDELAY

uint8\_t JT\_OFF : 4; // 0x0b.4, s:4 d: ONDELAY

uint8\_t JT\_ONDELAY : 4; // 0x0c.0, s:4 d: ONDELAY

uint8\_t JT\_OFFDELAY : 4; // 0x0c.4, s:4 d: ONDELAY

uint8\_t JT\_RAMPON : 4; // 0x0d.0, s:4 d: ONDELAY

uint8\_t JT\_RAMPOFF : 4; // 0x0d.4, s:4 d: ONDELAY

uint8\_t : 5; // 0x0e.0, s:5

uint8\_t OFFDELAY\_BLINK : 1; // 0x0e.5, s:1 d: ON

uint8\_t ON\_LEVEL\_PRIO : 1; // 0x0e.6, s:1 d: HIGH

uint8\_t ONDELAY\_MODE : 1; // 0x0e.7, s:1 d: SET\_TO\_OFF

uint8\_t OFF\_LEVEL : 8; // 0x0f.0, s:8 d: 0.0 %

uint8\_t ON\_MIN\_LEVEL : 8; // 0x10.0, s:8 d: 0.1 %

uint8\_t ON\_LEVEL : 8; // 0x11.0, s:8 d: 1.0 %

uint8\_t RAMP\_START\_STEP : 8; // 0x12.0, s:8 d: 0.05 %

uint8\_t RAMPON\_TIME : 8; // 0x13.0, s:8 d: 0 s

uint8\_t RAMPOFF\_TIME : 8; // 0x14.0, s:8 d: 0 s

uint8\_t DIM\_MIN\_LEVEL : 8; // 0x15.0, s:8 d: 0.0 %

uint8\_t DIM\_MAX\_LEVEL : 8; // 0x16.0, s:8 d: 1.0 %

uint8\_t DIM\_STEP : 8; // 0x17.0, s:8 d: 0.0 %

uint8\_t OFFDELAY\_STEP : 8; // 0x18.0, s:8 d: 0.05 %

uint8\_t OFFDELAY\_NEWTIME : 8; // 0x19.0, s:8 d: 0.5 s

uint8\_t OFFDELAY\_OLDTIME : 8; // 0x1a.0, s:8 d: 0.5 s

uint8\_t ELSE\_ACTION\_TYPE : 4; // 0x26.0, s:4 d: INACTIVE

uint8\_t : 1; // 0x26.4, s:2 d:

uint8\_t ELSE\_MULTIEXECUTE : 1;

uint8\_t ELSE\_OFF\_TIME\_MODE : 1; // 0x26.6, s:1 d: ABSOLUTE

uint8\_t ELSE\_ON\_TIME\_MODE : 1; // 0x26.7, s:1 d: ABSOLUTE

uint8\_t ELSE\_JT\_ON : 4; // 0x27.0, s:4 d: ONDELAY

uint8\_t ELSE\_JT\_OFF : 4; // 0x27.4, s:4 d: ONDELAY

uint8\_t ELSE\_JT\_ONDELAY : 4; // 0x28.0, s:4 d: ONDELAY

uint8\_t ELSE\_JT\_OFFDELAY : 4; // 0x28.4, s:4 d: ONDELAY

uint8\_t ELSE\_JT\_RAMPON : 4; // 0x29.0, s:4 d: ONDELAY

uint8\_t ELSE\_JT\_RAMPOFF : 4; // 0x29.4, s:4 d: ONDELAY

} \*l3; // 30 byte

enum E : uint8\_t { NO\_JUMP\_IGNORE\_COMMAND = 0x00, ONDELAY = 0x01, RAMPON = 0x02, ON = 0x03, OFFDELAY = 0x04, RAMPOFF = 0x05, OFF = 0x06 };

uint8\_t CT\_RAMPON : 4; RAMPON = 0x02

uint8\_t CT\_RAMPOFF : 4; RAMPOFF = 0x05

uint8\_t CT\_ONDELAY : 4; ONDELAY = 0x01

uint8\_t CT\_OFFDELAY : 4; OFFDELAY = 0x04

uint8\_t CT\_ON : 4; ON = 0x03

uint8\_t CT\_OFF : 4; OFF = 0x06

uint8\_t JT\_ON : 4; ON = 0x03

uint8\_t JT\_OFF : 4; OFF = 0x06

uint8\_t JT\_ONDELAY : 4; ONDELAY = 0x01

uint8\_t JT\_OFFDELAY : 4; OFFDELAY = 0x04

uint8\_t JT\_RAMPON : 4; RAMPON = 0x02

uint8\_t JT\_RAMPOFF : 4; RAMPOFF = 0x05

**Sensor Message received**

Condition table to check if the condition fit

Get the appropriate rule out of the condition table on base of the current state of the state machine:

uint8\_t CT\_RAMPON : 4; RAMPON = 0x02

uint8\_t CT\_RAMPOFF : 4; RAMPOFF = 0x05

uint8\_t CT\_ONDELAY : 4; ONDELAY = 0x01

uint8\_t CT\_OFFDELAY : 4; OFFDELAY = 0x04

uint8\_t CT\_ON : 4; ON = 0x03

uint8\_t CT\_OFF : 4; OFF = 0x06

Check the rule against the appropriate value:

uint8\_t COND\_VALUE\_LO : 8; // 0x04.0, s:8 d:

uint8\_t COND\_VALUE\_HI : 8; // 0x05.0, s:8 d:

Based on the result – true or false choose the appropriate action type register

uint8\_t ACTION\_TYPE : 4; // 0x0a.0, s:4 d: INACTIVE

uint8\_t ELSE\_ACTION\_TYPE : 4; // 0x26.0, s:4 d: INACTIVE

Valid values within the action type register

namespace DM\_ACTION {

enum E : uint8\_t { INACTIVE, JUMP\_TO\_TARGET, TOGGLE\_TO\_COUNTER, TOGGLE\_INV\_TO\_COUNTER, UPDIM, DOWNDIM, TOOGLEDIM, TOGGLEDIM\_TO\_COUNTER, TOGGLEDIM\_INVERS\_TO\_COUNTER, };

};

Call the respective function based on the action type register.

Valid Action Type‘s are:

INACTIVE:

Nothing to do

JUMP\_TO\_TARGET:

TOGGLE\_TO\_COUNTER:

On/Off on base of the last bit in the remote/sensor counter – 0 = Off, 1 = On

uint8\_t OFF\_LEVEL : 8; // 0x0f.0, s:8 d: 0.0 %

uint8\_t ON\_LEVEL : 8; // 0x11.0, s:8 d: 1.0 %

Set state machine to On or Off

TOGGLE\_INV\_TO\_COUNTER:

Same as TOGGLE\_TO\_COUNTER, but invers – 0 = On, 1 = Off

Set state machine to On or Off

UPDIM:

Increase dimmer value with

uint8\_t DIM\_STEP : 8; // 0x17.0, s:8 d: 0.0 %

Stop at

uint8\_t DIM\_MAX\_LEVEL : 8; // 0x16.0, s:8 d: 1.0 %

Set state machine to On

DOWNDIM:

Decrease dimmer value with

uint8\_t DIM\_STEP : 8; // 0x17.0, s:8 d: 0.0 %

Stop at

uint8\_t DIM\_MIN\_LEVEL : 8; // 0x15.0, s:8 d: 0.0 %

If dimmer value below

uint8\_t ON\_MIN\_LEVEL : 8; // 0x10.0, s:8 d: 0.1 %

set state machine to Off, otherwise to On

TOOGLEDIM:

Increase or decrease as opposite of the last actor change by calling UPDIM or DOWNDIM

TOGGLEDIM\_TO\_COUNTER:

Change dimmer value by

uint8\_t DIM\_STEP : 8; // 0x17.0, s:8 d: 0.0 %

in direction of the last bit of the remote/sensor counter value – 0 = DOWNDIM, 1 = UPDIM

TOGGLEDIM\_INVERS\_TO\_COUNTER:

Same as TOGGLEDIM\_TO\_COUNTER, but inverse 0 = UPDIM, 1 = DOWNDIM

Fernbedienungen senden bei langem Tastendruck in kurzen Abständen Telegramme mit gleichem Ereigniszähler. Im Aktor kann mit LONG\_MULTIEXECUTE ausgewählt werden, ob jedes dieser Telegramme ausgeführt werden soll oder ob jeder lange Tastendruck nur genau einmal ausgeführt wird. Das mehrfache Ausführen ist z. B. für das manuelle Dimmen nötig oder um z. B. mit einem Schaltaktor eine Türöffnerfunktion zu realisieren, die den Öffner nur so lange betätigt, wie auch die Fernbedienungstaste betätigt wird.

INACTIVE

Das eingetroffene Ereignis wird ignoriert und führt zu keiner Reaktion des Aktors.

JUMP\_TO\_TARGET

Abhängig vom aktuellen Zustand des Aktors wechselt dieser zum im zugehörigen Sprungziel (Jump-Target = JT) eingetragenen Profilabschnitt.

TOGGLE\_TO\_COUNTER

Der Aktor betrachtet das letzte Bit eines vom Sensor mitgesendeten Ereigniszählers. Ist dieses Bit 1, wird eingeschaltet und andernfalls ausgeschaltet.

TOGGLE\_INVERS\_TO\_COUNTER

Der Aktor schaltet genau entgegengesetzt zu TOGGLE\_TO\_COUNTER.

UPDIM

Es wird um eine Helligkeitsstufe hochgedimmt. Die Schrittweite und der Maximalwert lassen sich in anderen Parametern einstellen. (…\_DIM\_STEP und …\_DIM\_MAX\_LEVEL)

DOWNDIM

Es wird um eine Helligkeitsstufe runtergedimmt. Die Schrittweite und der Minimalwert lassen sich in anderen Parametern einstellen. (…\_DIM\_STEP und …\_DIM\_MIN\_LEVEL)

TOGGLEDIM

Es wird um eine Helligkeitsstufe hoch- oder runtergedimmt. Die Richtung kehrt sich dabei bei jeder Bedienung um. Dies ist ideal, wenn mehrere 1-Tasten-Sender einen Dimmer steuern sollen.

TOGGLEDIM\_TO\_COUNTER

Es wird um eine Helligkeitsstufe hoch- oder runtergedimmt. Die Richtung entspricht dabei dem letzten Bit des vom Sensor mitgesendeten Ereigniszählers. Dies ist ideal, wenn ein 1-Tasten-Sender mehrere Dimmer steuern soll.

TOGGLEDIM\_INVERS\_TO\_COUNTER

Der Aktor dimmt genau entgegengesetzt zu TOGGLEDIM\_TO\_COUNTER.

Ist als Aktion JUMP\_TO\_TARGET gewählt worden, prüft der Aktor nun, in welchem Profilabschnitt er sich aktuell befindet und in welchen er gegebenenfalls springen muss. Dies wird für jeden Abschnitt getrennt über die Parameter …\_JT\_… (Sprungziel = Jump Target) definiert (Bild 6):

NO\_JUMP\_IGNORE\_COMMAND

Es wird kein Sprung ausgeführt und laufende Timer werden nicht beeinflusst.

ONDELAY

Es wird zum Beginn der Einschaltverzögerung gesprungen. Mit ONDELAY\_MODE kann noch gewählt werden, ob der Aktor in dieser Zeit explizit ausschalten soll oder ob er seinen aktuellen Wert einfach behält.

RAMPON

Es wird zum Beginn der Einschaltrampe gesprungen. Mit …\_RAMP\_START\_STEP lässt sich ein zusätzlicher Pegelsprung als optische Rückmeldung z. B. bei sehr langsamen Rampen definieren.

ON

Es wird zum Beginn der Einschaltdauer gesprungen.

OFFDELAY

Es wird zum Beginn der Ausschaltverzögerung gesprungen. Mit …\_OFFDELAY\_STEP kann ein Pegelsprung zur Signalisierung der Ausschaltverzögerung konfiguriert werden. Aktiviert man auch noch …\_OFFDELAY\_BLINK, wird diese Verzögerungszeit durch ein Blinken signalisiert, dessen Puls-Pause-Verhältnis und Frequenz über die Parameter …\_OFFDELAY\_OLDTIME und …\_OFFDELAY\_NEWTIME definiert wird.

RAMPOFF

Es wird zum Beginn der Ausschaltrampe gesprungen. Mit …\_RAMP\_START\_STEP lässt sich ein zusätzlicher Pegelsprung als optische Rückmeldung z. B. bei sehr langsamen Rampen definieren.

OFF

Es wird zum Beginn der Ausschaltdauer gesprungen.

virtuelle Kanäle

Einige Dimmer unterstützen virtuelle Kanäle. Eigentlich haben sie keinen Hauptkanal sondern nur 3 virtuelle Kanäle, die je nach Verknüpfungslogik, den endgültigen Zustand des Aktors bestimmen. Dieses Verhalten wird in jedem Kanal per Register logicCombination gesetzt.

Folgende Logik wird dabei angewendet: phyLevel = (((0% o Ch1) o Ch2) o Ch3) [o = Logik-Platzhalter]

D.h. es wird immer der Kanal mit dem gesetzten Register mit dem vorherigen Teilergebnis kombiniert, angefangen mit Kanal1.

Es gibt folgende Logik-Optionen: [einzelne Betrachtung von (Vorergebnis o Kanal)]

Wert Bedeutung Beispiel (alle Werte in %)

inactive keine Verknüpfung 30 inactive 80 = 30

or oder (höchster Wert zählt) 30 or 80 = 80

and und (kleinster Wert zählt) 30 and 80 = 30

xor entweder-oder (nur 1 Kanal darf über 0% sein) 30 xor 80 = 0 aber 30 xor 0 = 30

nor oder negiert (höchster Wert subtrahiert von 100%) 30 nor 80 = !80 = 20

nand und negiert (niedrigster Wert subtrahiert von 100%) 30 nand 80 = !30 = 70

orinv oder mit invertiertem Kanal 30 orinv 80 = 30 or 20 = 30

andinv und mit invertiertem Kanal 30 andinv 80 = 30 and 20 = 20

plus Vorergebnis addiert mit Kanal 30 plus 80 = 100

minus Vorergebnis subtrahiert mit Kanal 30 minus 80 = 0

mul Vorergebnis multipliziert mit Kanal 30 mul 80 = 24

plusinv Vorergebnis addiert mit invertiertem Kanal 30 plusinv 80 = 30 plus 20 = 50

minusinv Vorergebnis subtrahiert mit invertiertem Kanal 30 minusinv 80 = 30 minus 20 = 10

mulinv Vorergebnis multipliziert mit invertiertem Kanal 30 mulinv 80 = 30 mul 20 = 6

invPlus invertiertes Vorergebnis addiert mit Kanal 30 invPlus 80 = 70 plus 80 = 100

invMinus invertiertes Vorergebnis subtrahiert mit Kanal 30 invMinus 80 = 70 minus 80 = 0

invMul invertiertes Vorergebnis multipliziert mit Kanal 30 invMul 80 = 70 mul 80 = 56

Elektronikwissen zu: Virtuelle Homematic Aktorkanäle und ihre Verknüpfungslogik

Die bisherigen Homematic Aktoren lassen sich meist nur über Programme für Vorrangsteuerungen oder in Abhängigkeit von Bedingungen schalten. Das verursacht dabei einerseits zeitliche Verzögerungen, andererseits ist die Funktion nur bei Verfügbarkeit der CCU gegeben. Könnten einfache logische Verknüpfungen und Bedingungen im Aktor verarbeitet werden, wäre dies zuverlässiger, schneller in der Reaktion und einfacher in der Programmierung. Hierzu wurden jetzt virtuelle Aktorkanäle und deren logische Verknüpfbarkeit bei Homematic eingeführt.

Die beiden neuen Homematic Dimmer HM-LC-Dim1PWM-CV und HM-LC-Dim1TPBU-FM sind die ersten Aktoren, die über dieses ganz spezielle neue Feature verfügen. Die Aktoren sind mit 3 virtuellen Aktorkanälen ausgestattet, deren Ausgangspegel über konfigurierbare Verknüpfungsregeln auf den realen physikalischen Ausgang ausgegeben werden. Im Grundzustand ist nur der erste virtuelle Kanal aktiv und das Gerät verhält sich wie ein üblicher 1-Kanal-Dimmer (Bild A). Aktiviert man jedoch über die CCU auch die anderen virtuellen Kanäle durch Definieren einer Verknüpfungsregel, so lassen sich die unterschiedlichsten Verhaltensweisen realisieren. Zuerst soll jedoch das Grundprinzip dieser Verknüpfungsregeln erläutert werden.

Für jeden der 3 virtuellen Kanäle kann als Verknüpfungsregel eine dieser Möglichkeiten gewählt werden:

Kanal inaktiv: Der Kanal wird bei der Verknüpfung ignoriert.

OR: Das Verknüpfungsergebnis ist der höhere von beiden Pegeln.

AND: Das Verknüpfungsergebnis ist der niedrigere von beiden Pegeln.

XOR: Ist nur einer der Pegel größer als 0 %, ist dieser Pegel auch das Verknüpfungsergebnis. In den anderen Fällen ist das Verknüpfungsergebnis 0 %.

NOR: Es wird die Verknüpfung OR ausgeführt und das Ergebnis anschließend invertiert (100 % - Pegel).

NAND: Es wird die Verknüpfung AND ausgeführt und das Ergebnis anschließend invertiert (100 % - Pegel).

OR\_INVERS: Der zu verknüpfende Kanal (rechts vom „o“) wird zuerst invertiert (100 % - Pegel) und anschließend die Verknüpfung OR ausgeführt.

AND\_INVERS: Der zu verknüpfende Kanal (rechts vom „o“) wird zuerst invertiert (100 % - Pegel) und anschließend die Verknüpfung AND ausgeführt.

PLUS: Die beiden Pegel werden addiert (max. 100 %).

MINUS: Die beiden Pegel werden subtrahiert (min. 0 %).

MULTI: Die beiden Pegel werden multipliziert.

PLUS\_INVERS: Der zu verknüpfende Kanal (rechts vom „o“) wird zuerst invertiert (100 % - Pegel) und anschließend die Verknüpfung PLUS ausgeführt.

MINUS\_INVERS: Der zu verknüpfende Kanal (rechts vom „o“) wird zuerst invertiert (100 % - Pegel) und anschließend die Verknüpfung MINUS ausgeführt.

MULTI\_INVERS: Der zu verknüpfende Kanal (rechts vom „o“) wird zuerst invertiert (100 % - Pegel) und anschließend die Verknüpfung MULTI ausgeführt.

INVERS\_PLUS: Die beiden Pegel werden addiert (max. 100 %) und das Ergebnis anschließend invertiert (100 % - Pegel).

INVERS\_MINUS: Die beiden Pegel werden subtrahiert (min. 0 %) und das Ergebnis anschließend invertiert (100 % - Pegel).

INVERS\_MULTI: Die beiden Pegel werden multipliziert und das Ergebnis anschließend invertiert (100 % - Pegel).

Die Verknüpfung der virtuellen Kanäle erfolgt dann nach folgendem Schema, wobei die kleinen Kreise die Verknüpfungszeichen und A bis C die Pegel der Kanäle 1 bis 3 darstellen: Ausgangspegel = ([(0 % o A) o B] o C) Die genaue Abarbeitung der Formel sieht dabei so aus: Zuerst wird der Pegel 0 % mit dem Pegel von Kanal 1 (A) nach seiner Regel verknüpft. Das Ergebnis wird danach mit B entsprechend der zugehörigen Regel verknüpft. Zuletzt wird das bisherige Ergebnis noch mit C nach der Regel von Kanal 3 verknüpft. Der Default-Wert des Verknüpfungsparameters bei Kanal 1 ist „OR“ und bei allen anderen „Kanal inaktiv“, damit die zusätzlichen virtuellen Kanäle nicht verwirren, wenn nur die Grundfunktion der Geräte benötigt wird. Die Werte für A, B und C entsprechen dabei dem Pegel, der aus dem Aktorprofil an die Endstufenansteuerung übergeben würde. Befindet sich ein Dimmer also z. B. in der Ausschaltvorwarnung mit aktivem Blinken, so ist dieses Blinken auch in dem Pegel von z. B. Kanal 1 zu finden.

An einem Beispiel soll die Wirkungsweise verdeutlicht werden:

Kanal 1: OR, A = 70 %

Kanal 2: NOR, B = 25 %

Kanal 3: AND\_INVERS, C = 80 %

Ausgangspegel = ([(0 % OR 70 %) NOR 25 %] AND\_INVERS 80 %)

0 % OR 70 % = 70 %

70 % NOR 25 % = 30 %

30 % AND\_INVERS 80 % = 20 %

Der Ausgangspegel ergibt hier 20 %.

Zuletzt sollen noch zwei praktische Anwendungsbeispiele die Vorteile der virtuellen Kanäle und der Verknüpfungsregeln aufzeigen.

1. Eine Flurbeleuchtung soll mit Bewegungsmelder, Tag-/Nacht-Information und Wandtaster gesteuert werden: Der Bewegungsmelder wird mit Kanal 1 verknüpft und als Aktion ein Einschalten auf 80 % für 60 Sekunden festgelegt. Die Verknüpfungsregel für diesen Kanal bleibt auf dem Default-Wert „OR“. Für die Tag-/Nacht-Information schaltet die CCU 1 den Kanal 2 bei Nacht auf 100 % und am Tag auf 0 %. Die Verknüpfungsregel für diesen Kanal wird auf „AND“ gesetzt. Der Wandtaster wird mit Kanal 3 verknüpft und als Aktion das Aus-/Einschalten mit einem EIN-Pegel von 100 % eingestellt. Als Verknüpfungsregel für Kanal 3 wird wieder „OR“ gewählt. Damit ergibt sich ein sehr praxisgerechtes Verhalten. Das Licht im Flur geht nachts bei Bewegung kurz in reduzierter Helligkeit an. Tagsüber geht das Licht im Flur nicht automatisch an. Auf Tastendruck kann das Flurlicht jederzeit dauerhaft auf maximaler Helligkeit eingeschaltet werden. Beendet man diese Dauerbeleuchtungsphase durch einen weiteren Tastendruck, steht man nachts nicht im Dunkeln, da das durch den Bewegungsmelder ausgelöste Profil mit der dort eingestellten Helligkeit nicht ausgeschaltet wird. Dieses Szenario lässt sich noch weiter perfektionieren, indem die Zentrale den Kanal 2 z. B. von 23:00 bis 06:00 Uhr von 100 % auf 30 % absenkt. Damit ist die normale Bewegungsmelder- Helligkeit von 80 % für die Schlafphase auf 30 % reduziert und man wird bei einem nächtlichen „Ausflug“ nicht geblendet. Ein großer Vorteil dieser Realisierung ist das schnelle Ansprechen der Beleuchtung auf Bewegungsmelder-Impulse, da eine direkte Kommunikation erfolgt und kein Umweg über die Zentrale gemacht werden muss. Ein weiterer Vorteil ist die einfache Vorrangsteuerung des Wandtasters und die einfache tageszeitabhängige Änderung der Beleuchtungsstärke. Um ein ähnliches Verhalten nur über Programme auf der CCU 1 abzubilden, wären bereits sehr aufwendige Programme nötig, wobei zusätzlich zeitliche Verzögerungen und deutlich mehr Funkverkehr erzeugt würden.

2. Die Deckenlampe im Wohnzimmer soll automatisch abgedunkelt werden, wenn die Leselampe eingeschaltet wird: Wenn vorher schon eine Fernbedienung so an die Dimmer für die Deckenlampe und die Leselampe angelernt ist, dass diese mit verschiedenen Tasten separat gesteuert werden können, muss nur noch der Kanal 2 des Deckenlampen-Dimmers zusätzlich mit den Fernbedienungstasten verknüpft werden, die primär die Leselampe steuern. Als Verknüpfungsregel für Kanal 2 wird dabei „MINUS“ gewählt. Um zu verhindern, dass die Deckenlampe zu stark abgedunkelt wird, sollten der EIN-Pegel und die Obergrenze für das Aufdimmen von Kanal 2 in den Easy-Profil-Einstellungen (siehe Bild B) oder den Experten-Einstellungen (siehe Bild C) z. B. auf 50 % gesetzt werden. Wird bei eingeschalteter Deckenlampe nun die Leselampe eingeschaltet, reduziert sich die Helligkeit der Deckenlampe um 50 Prozentpunkte. Schaltet man die Leselampe wieder aus, geht die Deckenlampe auf ihre alte Helligkeit zurück. Die Verknüpfungsregel MINUS hat in dieser Anwendung eventuell den Nachteil, dass der eingestellte Helligkeitswert der Deckenlampe keinen Einfluss darauf hat, um wie viel Prozentpunkte ihre Helligkeit abgesenkt wird. Soll die Absenkung um einen prozentualen Anteil von der aktuell eingestellten Helligkeit erfolgen, ist für Kanal 2 die Verknüpfungsregel MULTI\_INVERS zu wählen. Wird die Leselampe dann beispielsweise auf 80 % eingeschaltet, wird die Deckenlampe auf 20 % ihrer vorherigen Helligkeit reduziert.